

## WHAT IS CLAIMED IS:

1. A system for loading an executable image on to at least one image receiver, said system comprising:

at least one image source, said image source having access to at least one executable image; and

at least one image receiver coupled to said at least one image source by a distributed interconnect;

wherein said at least one image source is capable of communicating said executable image to said at least one image receiver across said distributed interconnect for loading on to said at least one image receiver.

2. The system of claim 1, wherein said distributed interconnect comprises a switch fabric.

3. The system of claim 1, wherein said distributed interconnect comprises a virtual distributed interconnect.

4. The system of claim 2, wherein said executable image comprises a diagnostic image.

5. The system of claim 2, wherein said executable image comprises an initial image.

6. The system of claim 5, wherein said image source comprises a management processing engine, and wherein said image receiver comprises an application processing engine.

7. The system of claim 6, wherein said initial image comprises at least one of a boot code, an operating system, an application program interface, an application, , or a combination thereof.

8. The system of claim 6, wherein said image source and said image receiver comprise components of an information management system.

9. The system of claim 6, wherein said information management system comprises multiple image sources, multiple image receivers, or a combination thereof; and wherein said multiple image sources are coupled to at least one image receiver by said switch fabric, wherein said multiple image receivers are coupled to at least one image source by said switch fabric, or a combination thereof.

10. The system of claim 8, wherein said information management system comprises a content delivery system.

11. The system of claim 10, wherein said content delivery system comprises an endpoint content delivery system.

12. The system of claim 2, wherein said image source has access to a plurality of different executable images; wherein said image receiver comprises a first image receiver; and wherein said method comprises selecting and communicating a first one of said plurality of executable images from said image source to said first image receiver across said switch fabric.

13. The system of claim 12, wherein said method further comprises selecting and communicating a second one of said plurality of executable images from said image source across said switch fabric to a second image receiver coupled to said image source by said switch fabric.

14. The system of claim 2, wherein a first image source has access to a first executable image, and a second image source has access to a second executable image, said first and second executable images being different from each other; wherein said first and second image sources are coupled to said at least one image receiver by said switch fabric; and wherein said method comprises selecting and communicating at least one of said first or second executable images from said respective first or second image source to said image receiver across said switch fabric.

15. A method of loading an executable image on to at least one image receiver, said method comprising:

communicating said executable image from at least one image source to said at least one image receiver; and

loading said executable image on to said image receiver;

wherein said at least one image source and said at least one image receiver are coupled together by a distributed interconnect; and wherein said executable image is communicated from said at least one image source to said at least one image receiver across said distributed interconnect.

16. The method of claim 15, wherein said distributed interconnect comprises a switch fabric.

17. The method of claim 15, wherein said distributed interconnect comprises a virtual distributed interconnect.

5

18. The method of claim 16, wherein said executable image comprises a diagnostic image.

10

19. The method of claim 16, wherein said executable image comprises an initial image.

20. The method of claim 19, wherein said image source comprises a management processing engine, and wherein said image receiver comprises an application processing engine.

21. The method of claim 20, wherein said initial image comprises at least one of a boot code, an operating system, an application program interface, an application, or a combination thereof.

22. The method of claim 20, wherein said image source and said image receiver comprise components of an information management system.

23. The method of claim 22, wherein said information management system comprises a content delivery system.

24. The method of claim 23, wherein said content delivery system comprises an endpoint content delivery system.

25. The method of claim 16, wherein said executable image remains quiescent after said loading on said image receiver; and wherein said method further comprises communicating an execution signal to said image receiver across said switch fabric, said execution signal instructing said image receiver to begin execution of said executable image.

26. The method of claim 16, wherein said image source has access to a plurality of different executable images; wherein said image receiver comprises a first image receiver; and wherein said method comprises selecting and communicating a first one of said plurality of executable images from said image source to said first image receiver across said switch fabric.

27. The method of claim 26, wherein said method further comprises selecting and communicating a second one of said plurality of executable images from said image source across said switch fabric to a second image receiver coupled to said image source by said switch fabric.

28. The method of claim 16, wherein a first image source has access to a first executable image, and a second image source has access to a second executable image, said first and second executable images being different from each other; wherein said first and second image sources are coupled to said at least one image receiver by said switch fabric; and wherein said method comprises selecting and communicating at least one of said first or second executable images from said respective first or second image source to said image receiver across said switch fabric.

29. A system for interfacing a first processing object with a second processing object, said system comprising:

a first processing engine, said first processing engine having said first processing object residing thereon; and

a second processing engine coupled to said first processing engine by a distributed interconnect, said second processing engine having said second processing object residing thereon;

5

wherein said second processing object is specific to said first processing object, and wherein said first object is capable of interfacing with said second object across said distributed interconnect.

10

30. The system of claim 29, wherein said distributed interconnect comprises a switch fabric.

31. The system of claim 29, wherein said distributed interconnect comprises a virtual distributed interconnect.

32. The system of claim 30, wherein said interfacing comprises accessing, managing, or a combination thereof.

33. The system of claim 32, wherein said first processing engine comprises an application processing engine; and wherein said second processing comprises a storage processing engine.

34. The system of claim 33, wherein said first processing object comprises an application object; and wherein said second processing object comprises a buffer/cache object that is specific to said application object.

35. The system of claim 33, wherein said first processing object comprises a file system object; and wherein said second processing object comprises a logical volume management object that is specific to said file system object.

5

36. The system of claim 34, wherein said first processing engine further has a file system object residing thereon, and said second processing engine further has a logical volume management object residing thereon; wherein said logical volume management object is specific to said file system object; and wherein said file system object is capable of interfacing with said logical volume management object across said distributed interconnect.

10

37. The system of claim 36, further comprising at least one content source coupled to said storage processing engine; and wherein said second processing engine is capable of providing said first processing engine with access to content available from said content source across said distributed interconnect.

15

38. The system of claim 33, wherein said first processing engine and said second processing engine comprise components of an information management system.

20

39. The system of claim 38 wherein said information management system comprises a content delivery system.

25

40. The system of claim 39, wherein said content delivery system comprises an endpoint content delivery system.

30

41. The system of claim 30, wherein said interfacing comprises managing, and wherein said managing occurs over said distributed interconnect via a separate designated communication path.

42. The system of claim 30, wherein said system comprises:

at least two first processing engines, each of said first processing engines having at  
least one respective first processing object residing thereon;

at least two second processing engines, each of said second processing engines having  
at least one respective second processing object residing thereon;

wherein said first objects are capable of interfacing with said second objects across  
said distributed interconnect;

wherein the characteristics of a given second processing object residing on at least  
one of said second processing engines differs from the characteristics of an other second  
processing object residing on an other one of said second processing engines; and

wherein said given second processing object is specific to a given first processing  
object residing on at least one of said first processing engines, and wherein said other second  
processing object is specific to an other first processing object residing on at least one of said  
first processing engines.

43. The system of claim 42, wherein any one of said first processing engines is  
selectably interconnectable to any one of said second processing engines across said  
distributed interconnect so that a selected first processing object residing on one of said first  
processing engines may be selectably interfaced with a selected second processing object  
residing on one of said second processing engines that is specific to said selected first  
processing object.

44. The system of claim 43, wherein any one of said first processing engines is  
selectably interconnectable to any one of said second processing engines across said  
distributed interconnect so that a selected first processing object residing on one of said first



processing engines may be selectably interfaced with a selected second processing object residing on one of said second processing engines on a dynamic basis.

5           45.    The system of claim 44, wherein said selected first processing object is selectably interfaceable with said second processing object in response to a first request for information management, relative to selective interfacing operations between first processing objects and second processing objects in response to a second request for information management, in a manner based at least in part on one or more parameters associated with  
10 individual respective requests for information management.

          46.    The system of claim 43, wherein each of said first processing engines comprises a first application processing engine of a content delivery system, and wherein  
5 each of said second processing engines comprises a storage processing engine of said content delivery system; wherein said selected first processing object may be selectably interfaced with said selected second processing object to allow a first processing engine on which said selected first processing object resides to retrieve content from a content source using a second processing engine on which said selected second processing object resides.

          47.    The system of claim 46, wherein said selected first processing object comprises a selected application processing object and wherein said selected second  
20 processing object comprises a selected buffer/cache processing object specific to said selected application processing object; wherein said selected first processing object comprises a selected file system processing object and wherein said selected second processing object comprises a selected logical volume management processing object specific to said selected file system processing object; or a combination thereof.

          48.    The system of claim 47, wherein said content delivery system comprises an endpoint content delivery system.

49. A method of interfacing a first processing object with a second processing object, said method comprising interfacing said second processing object with said first processing object across a distributed interconnect; wherein said second processing object is specific to said first processing object.

5

50. The method of claim 49, wherein said first processing object resides on a first processing engine; wherein said second processing object resides on a second processing engine; and wherein said interfacing comprises coupling said first processing engine to said second processing engine using said distributed interconnect.

10

51. The method of claim 50, wherein said distributed interconnect comprises a switch fabric.

5

52. The method of claim 50, wherein said distributed interconnect comprises a virtual distributed interconnect.

20

53. The method of claim 51, wherein said interfacing comprises accessing, managing, or a combination thereof.

25

54. The method of claim 53, wherein said first processing engine comprises an application processing engine; and wherein said second processing comprises a storage processing engine.

30

55. The method of claim 54, wherein said first processing object comprises an application object; and wherein said second processing object comprises a buffer/cache object that is specific to said application object.

56. The method of claim 54, wherein said first processing object comprises a file system object; and wherein said second processing object comprises a logical volume management object that is specific to said file system object.

57. The method of claim 55, wherein said first processing engine further has a file system object residing thereon, and said second processing engine further has a logical volume management object residing thereon; wherein said logical volume management object is specific to said file system object; and wherein said method further comprises interfacing said logical volume management object with said file system object across said distributed interconnect.

58. The method of claim 57, wherein said storage processing engine is coupled to at least one content source; and wherein said method further comprises providing using said storage processing engine to provide said first processing engine with access to content available from said content source across said distributed interconnect.

59. The method of claim 54, wherein said first processing engine and said second processing engine comprise components of an information management system.

60. The method of claim 59, wherein said information management system comprises a content delivery system.

61. The method of claim 60, wherein said content delivery system comprises an endpoint content delivery system.

62. The method of claim 51, wherein said interfacing comprises managing, and wherein said managing occurs over said distributed interconnect via a separate designated communication path.

63. The method of claim 51, wherein:

5 said first processing engine and said second processing engine comprise part of a system having at least two first processing engines and at least two second processing engines, each of said first processing engines having at least one respective first processing object residing thereon, and each of said second processing engines having at least one respective second processing object residing thereon;

10 wherein the characteristics of a given second processing object residing on at least one of said second processing engines differs from the characteristics of an other second processing object residing on an other one of said second processing engines;

15 wherein said given second processing object is specific to a given first processing object residing on at least one of said first processing engines, and wherein said other second processing object is specific to an other first processing object residing on at least one of said first processing engines; and

20 wherein said interfacing comprises interfacing a first processing object residing on one of said first processing engines with a second processing object residing on one or said second processing engines.

25  
64. The method of claim 63, wherein any one of said first processing engines is selectably interconnectable to any one of said second processing engines across said distributed interconnect, and wherein said interfacing comprises using said distributed interconnect to selectably interface a selected first processing object residing on one of said first processing engines with a selected second processing object residing on one of said second processing engines that is specific to said selected first processing object.

65. The method of claim 64, wherein said interfacing comprises selectably interfacing said selected first processing object with said second processing object on a dynamic basis.

66. The method of claim 65, wherein said method further comprises managing said selectable interfacing of said selected first processing object with said second processing object in response to a first request for information management, relative to selective interfacing operations between first processing objects and second processing objects in response to a second request for information management, in a manner based at least in part on one or more parameters associated with individual respective requests for information management.

67. The method of claim 64, wherein each of said first processing engines comprises a first application processing engine of a content delivery system, and wherein each of said second processing engines comprises a storage processing engine of said content delivery system; and wherein said method comprises selectably interfacing said selected first processing object with said selected second processing object to allow a first processing engine on which said selected first processing object resides to retrieve content from a content source using a second processing engine on which said selected second processing object resides.

68. The method of claim 67, wherein said selected first processing object comprises a selected application processing object and wherein said selected second processing object comprises a selected buffer/cache processing object specific to said selected application processing object; wherein said selected first processing object comprises a selected file system processing object and wherein said selected second processing object comprises a selected logical volume management processing object specific to said selected file system processing object; or a combination thereof.

69. The method of claim 68, wherein said content delivery system comprises an endpoint content delivery system.

5 70. A system for managing a processing object, said system comprising:

a first processing engine, said first processing engine having at least one first processing object residing thereon; and

10 a management entity coupled to said first processing engine by a distributed interconnect, said management entity capable of managing said first processing object residing on said first processing engine across said distributed interconnect.

15 71. The system of claim 70, wherein said distributed interconnect comprises a switch fabric.

20 72. The system of claim 70, wherein said distributed interconnect comprises a virtual distributed interconnect.

25 73. The system of claim 71, wherein said management entity comprises at least one of a separate processing engine, a separate system, a manual input, or a combination thereof.

30 74. The method of claim 73, wherein said management entity comprises a separate processing engine.

75. The system of claim 74, wherein said separate processing engine comprises a system management processing engine; wherein said first processing engine comprises a

storage processing engine; and wherein said at least one first processing object comprises a buffer cache algorithm, logical volume management algorithm, or a combination thereof.

5           76.     The system of claim 75, further comprising a second processing engine, said second processing engine being coupled to said first processing engine by said distributed interconnect; wherein said second processing engine has at least one second processing object residing thereon; wherein said first processing object is specific to said second processing object; and wherein said first processing object is capable of interfacing with said second processing object across said distributed interconnect.

10           77.     The system of claim 76, wherein said second processing engine comprises an application processing engine; wherein said second processing object comprises an application object; and wherein said first processing object comprises a buffer/cache object that is specific to said application object.

15           78.     The system of claim 76, wherein said second processing engine comprises an application processing engine; wherein said second processing object comprises a file system object; and wherein said first processing object comprises a logical volume management object that is specific to said file system object.

20           79.     The system of claim 76, wherein said second processing engine comprises an application processing engine; wherein said at least one second processing object comprises an application object and a file system object; and wherein said at least one first processing object comprises a buffer/cache object that is specific to said application object, and a logical volume management object that is specific to said file system object.

25           80.     The system of claim 77, wherein said first processing engine and said second processing engine comprise components of an information management system.

81. The system of claim 80, wherein said information management system comprises a content delivery system.

5

82. The system of claim 81, wherein said content delivery system comprises an endpoint content delivery system.

10

83. A method of managing at least one processing object, said method comprising managing said processing object across a distributed interconnect.

15

84. The method of claim 83, wherein said at least one processing object comprises a first processing object residing on a first processing engine; wherein said first processing engine is coupled to a management entity by said distributed interconnect; and wherein said managing comprises using said management entity to manage said first processing object across said distributed interconnect.

20

85. The method of claim 83, wherein said distributed interconnect comprises a switch fabric.

25

86. The method of claim 83, wherein said distributed interconnect comprises a virtual distributed interconnect.

30

87. The method of claim 85, wherein said management entity comprises at least one of a separate processing engine, a separate system, a manual input, or a combination thereof.



88. The method of method 87, wherein said management entity comprises a separate processing engine.

89. The method of claim 88, wherein said separate processing engine comprises a system management processing engine; wherein said first processing engine comprises a storage processing engine; and wherein said at least one first processing object comprises a buffer cache algorithm, logical volume management algorithm, or a combination thereof.

90. The method of claim 89, wherein a said second processing engine is coupled to said first processing engine by said distributed interconnect; wherein said second processing engine has at least one second processing object residing thereon; wherein said first processing object is specific to said second processing object; and wherein said method further comprises interfacing said first processing object with said second processing object across said distributed interconnect.

91. The method of claim 90, wherein said second processing engine comprises an application processing engine; wherein said second processing object comprises an application object; and wherein said first processing object comprises a buffer/cache object that is specific to said application object.

92. The method of claim 90, wherein said second processing engine comprises an application processing engine; wherein said second processing object comprises a file system object; and wherein said first processing object comprises a logical volume management object that is specific to said file system object.

93. The method of claim 90, wherein said second processing engine comprises an application processing engine; wherein said at least one second processing object comprises an application object and a file system object; and wherein said at least one first processing

object comprises a buffer/cache object that is specific to said application object, and a logical volume management object that is specific to said file system object.

94. The method of claim 91, wherein said first processing engine and said second processing engine comprise components of an information management system.

95. The method of claim 94, wherein said information management system comprises a content delivery system.

96. The method of claim 95, wherein said content delivery system comprises an endpoint content delivery system.

97. A method of coordinating a group of multiple processing engines in the performance of an operating task, said method comprising broadcasting a multicast message to said group of multiple processing engines across a distributed interconnect, said multicast facilitating the performance of said operating task.

98. The method of claim 97, wherein said distributed interconnect comprises a switch fabric.

99. The method of claim 97, wherein said distributed interconnect comprises a virtual distributed interconnect.

100. The method of claim 98, wherein said operating task comprises a failover operation, a load-balancing operation, a debugging operation, an operation to monitor a status of one or more information management resources, or a combination thereof.

101. The method of claim 100, wherein said operating task comprises a failover operation; and wherein said method comprises broadcasting said multicast message across said distributed interconnect to keep one or more of said group of processing engines apprised of the status of one or more individual members of said group of processing engines.

102. The method of claim 101, wherein said method comprises using said one or more individual members of said group of multiple processing engines to broadcast periodic multicast communications to other members of said group of multiple processing engines to indicate normal operations; and wherein said method further comprises implementing said failover operation upon absence of said periodic multicast communications from a failed processing engine by using another processing engine to assume the load or tasks of said failed processing engine.

103. The method of claim 101, wherein said method comprises broadcasting a multicast failure alarm from a failed processing engine of said group of multiple processing engines to other members of said group of multiple processing engines; and wherein said method further comprises implementing said failover operation upon broadcast of said multicast failure alarm by using another processing engine to assume the load or tasks of said failed processing engine.

104. The method of claim 101, wherein said method comprises:

using one or more designated members of said group of multiple processing engines to monitor and to detect failures of one or more other members of the group;

upon detection of a failed processing engine, using said one or more designated members of said group of multiple processing engines to broadcast a multicast failure alarm to other members of said group of multiple processing engines; and

wherein said method further comprises implementing said failover operation upon broadcast of said multicast failure alarm by using another processing engine to assume the load or tasks of said failed processing engine.

5

105. The method of claim 100, wherein said operating task comprises a load balancing operation; and wherein said method comprises broadcasting said multicast message across said distributed interconnect to keep said group of processing engines apprised of the status of one or more individual members of said group of processing engines.

10

106. The method of claim 105, wherein said method comprises using said one or more individual members of said group of multiple processing engines to broadcast multicast communications to other members of said group of multiple processing engines to indicate a workload level of said one or more individual members of said group of multiple processing engines; and wherein said method further comprises implementing said load balancing operation upon receipt of said multicast communications by transferring workload among two or more members of said group of multiple processing engines to balance workload level among said two or more members of said group of multiple processing engines.

15  
20

107. The method of claim 100, wherein said multicast message comprises a multicast query from a given member of said group of multiple processing engines, said multicast query requesting information from one or more other members of said group of processing engines; and wherein said method further comprises implementing a failover operation, a load balancing operation, or a combination thereof among two or more members of said group of multiple processing engines upon receipt of said requested information by transferring workload among two or more members of said group of multiple processing engines based at least in part on said requested information.

25

30

108. The method of claim 105, wherein said method comprises:

using one or more designated members of said group of multiple processing engines to monitor and to detect workload level of one or more other members of the group;

upon detection of a workload level imbalance among said one or more other members of the group, using said one or more designated members of said group of multiple processing engines to broadcast a multicast communication to other members of said group of multiple processing engines to indicate a workload level of said one or more individual members of said group of multiple processing engines; and

wherein said method further comprises implementing said load balancing operation upon receipt of said multicast communications by transferring workload among two or more members of said group of multiple processing engines to balance workload level among said two or more members of said group of multiple processing engines.

109. The method of claim 100, wherein said method comprises broadcasting said multicast message across said distributed interconnect to keep one or more of said group of processing engines apprised of one or more defined characteristics of one or more other members of said group of processing engines; wherein said defined characteristics comprise at least one of common processing characteristics, related processing characteristics, or a combination thereof.

110. The method of claim 109, wherein said method comprises using a given application running on one of said members of said group of processing engines to broadcast said multicast message; and wherein said multicast message comprises a multicast query for another instance of itself running on one or more other members of said group of processing engines.

111. The method of claim 109, wherein said method comprises using a given application running on one of said members of said group of processing engines to broadcast said multicast message; and wherein said multicast message comprises a multicast query for

services or other application on which said given application depends running on one or more other members of said group of processing engines.

5           112.    The method of claim 100, wherein said operating task comprises a debugging operation; wherein said multicast message comprises communications between two or more of said members of said group of multiple processing engines; and wherein said method comprises monitoring said multicast message using a given member of said group of multiple processing engines, and further comprising at least one of viewing, analyzing, or storing said  
10   multicast message on said given member of said group of multiple processing engines.

15           113.    The method of claim 112, further comprising making said multicast message accessible on said given member of said group of multiple processing engines for debug analysis by human operator, further external processing and debug analysis, or a combination thereof.

20           114.    The method of claim 112, further comprising performing debug analysis on said multicast message using said given member of said group of multiple processing engines.

25           115.    The method of claim 114, further comprising using said given member of said group of multiple processing engines to identify problems with said software code, to take corrective action to address problems with said software code, to report an external alarm upon identification of problems with said software code, or a combination thereof.

30           116.    The method of claim 114, wherein said two or more processing engines comprise an application processing engine, a storage processing engine, a transport processing engine, or a combination thereof; and wherein said given member of said group of multiple processing engines comprises a system management processing engine.

117. The method of claim 100, wherein said multiple processing engines comprise components of the same information management system, components of multiple information management systems, or a combination thereof.

5

118. The method of claim 100, wherein said multiple processing engines comprise components of a content delivery system.

10

119. The method of claim 118, wherein said content delivery system comprises an endpoint content delivery system.

120. A method of analyzing software code running on a first processing engine, said method comprising communicating debug information associated with said code from said first processing engine to a second processing engine across a distributed interconnect.

121. The method of claim 120, wherein said distributed interconnect comprises a switch fabric.

122. The method of claim 120, wherein said distributed interconnect comprises a virtual distributed interconnect.

123. The method of claim 121, further comprising at least one of viewing, analyzing, or storing said debug information on said second processing engine.

124. The method of claim 121, further comprising making said debug information accessible on said second processing engine for analysis by human operator, further external processing and analysis, or a combination thereof.

125. The method of claim 123, further comprising analyzing said debug information using said second processing engine.

5

126. The method of claim 123, wherein said distributed interconnect comprises a switch fabric; wherein said first processing engine comprises an application processing engine; and wherein said second processing engine comprises a system management processing engine.

10

127. The method of claim 123, wherein said distributed interconnect comprises a virtual distributed interconnect; wherein said first processing engine comprises an application processing functionality; wherein said second processing engine comprises a host processing functionality; wherein said application processing functionality and said host processing functionality are distributively interconnected across a network by said virtual distributed interconnect.

15

128. The method of claim 123, wherein said first and second processing engines comprise components of the same information management system, components of different information management systems, or a combination thereof.

20

129. The method of claim 123, wherein said first and second processing engines comprise components of a content delivery system.

25

130. The method of claim 129, wherein said content delivery system comprises an endpoint content delivery system.

30

131. A method of managing the manipulation of information among a group of multiple



processing engines in an information management environment, each of said processing engines being capable of performing one or more information manipulation tasks, said method comprising:

5 receiving first and second requests for information management;

selecting a first processing flow path among said group of processing engines in order to perform a first selected combination of information manipulation tasks associated with said first request for information management; and

10

selecting a second processing flow path among said group of processing engines in order to perform a second selected combination of information manipulation tasks associated with said second request for information management;

15

wherein said group of multiple processing engines are coupled together by a distributed interconnect, wherein said first processing flow path is different from said second processing flow path, and wherein said first and second processing flow paths are each selected using said distributed interconnect.

20

132. The method of claim 131, wherein said distributed interconnect comprises a switch fabric.

25

133. The method of claim 132, wherein each of said multiple processing engines is assigned separate information manipulation tasks in an asymmetrical multi-processor configuration.

30

134. The method of claim 133, wherein said distributed interconnect comprises a virtual distributed interconnect.

135. The method of claim 133, wherein said selecting of said first and second processing

flow paths is based at least in part on respective first and second parameters associated with each of said first and second requests for information management, based at least in part on respective first and second parameters associated with the respective particular type of information management requested by each of said first and second requests for information management, based at least in part on respective first and second parameters associated with particular user and/or class of users generating each of said first and second requests for information management, based at least in part on respective first and second parameters associated with system workload implicated by each of said first and second requests for information management, or a combination thereof.

136. The method of claim 131, wherein said selecting of said first processing flow path is based at least in part on a respective first parameter associated with said first request for information management; and wherein said selecting of said second processing flow path is based at least in part on a respective second parameter associated with said second request for information management.

137. The method of claim 136, wherein at least one of said first and second parameters comprises a priority-indicative parameter.

138. The method of claim 136, wherein at least one of said first and second parameters comprises a parameter indicative of one or more selectable information manipulation tasks; and wherein a respective first or second processing flow path selected based at least in part on said parameter indicative of one or more selectable information manipulation tasks comprises a processing flow path that includes one or more processing engines capable of performing said one or more selectable information manipulation tasks.

139. The method of claim 138, wherein one or more processing engines of said first processing flow path are capable of performing one or more of the same core information manipulation tasks as performed by one or more processing engines of said second processing flow path.

140. The method of claim 139, wherein said one or more selectable information manipulation tasks comprise at least one of data encryption, data compression, a security function, transcoding, content filtering, content transformation, filtering based on metadata, metadata transformation, or a combination thereof.

141. The method of claim 131, wherein one or more of said multiple processing engines is capable of recognizing one or more of said respective first and second parameters and is further capable of altering at least a portion of a processing flow path based upon said recognized parameter; and wherein said selecting of at least one of said first or said second processing flow paths comprises using said one or more of said multiple processing engines to recognize one or more of said respective first and second parameters and to alter at least a portion of at least one of said first or said second processing flow paths based at least in part upon said recognized parameter.

142. The method of claim 131, wherein said multiple processing engines comprise components of the same information management system, components of multiple information management systems, or a combination thereof.

143. The method of claim 132, wherein said multiple processing engines comprise components of a content delivery system; and wherein said information management comprises delivery of content.

144. The method of claim 143, wherein said content delivery system comprises an endpoint content delivery system.

145. The method of claim 143, wherein said selecting of said first and second processing flow paths is based at least in part on respective first and second parameters associated with

each of said first and second requests for content delivery, on respective first and second parameters associated with the respective particular type of content delivery requested by each of said first and second requests for information management, on respective first and second parameters associated with particular user and/or class of users generating each of said first and second requests for delivery, on respective first and second parameters associated with system workload implicated by each of said first and second requests for content delivery, or a combination thereof.

146. The method of claim 143, wherein said selecting of said first processing flow path is based at least in part on a respective first parameter associated with said first request for information management; and wherein said selecting of said second processing flow path is based at least in part on a respective second parameter associated with said second request for information management.

147. The method of claim 146, wherein at least one of said first and second parameters comprises a priority-indicative parameter.

148. The method of claim 147, wherein at least one of said first and second parameters comprises a parameter indicative of one or more selectable information manipulation tasks; and wherein a respective first or second processing flow path selected based at least in part on said parameter indicative of one or more selectable information manipulation tasks comprises a processing flow path that includes one or more processing engines capable of performing said one or more selectable information manipulation tasks.

149. The method of claim 148, wherein one or more processing engines of said first processing flow path are capable of performing one or more of the same core information manipulation tasks as performed by one or more processing engines of said second processing flow path.

150. The method of claim 149, wherein said same core information manipulation tasks comprise information manipulation tasks performed by at least one of a network application processing engine, a network transport processing engine, a storage management processing engine, a network interface processing engine, or a combination thereof.

5

151. The method of claim 149, wherein said one or more selectable information manipulation tasks comprise at least one of data encryption, data compression, a security function, transcoding, content filtering, content transformation, filtering based on metadata, metadata transformation, or a combination thereof.

10

152. The method of claim 145, wherein one or more of said multiple processing engines is capable of recognizing one or more of said respective first and second parameters and is further capable of altering at least a portion of a processing flow path based upon said recognized parameter; and wherein said selecting of at least one of said first or said second processing flow paths comprises using said one or more of said multiple processing engines to recognize one or more of said respective first and second parameters and to alter at least a portion of at least one of said first or said second processing flow paths based at least in part upon said recognized parameter.

15

20

153. The method of claim 152, wherein said recognized parameter comprises a substantive characteristic associated with requested content.

25

154. The method of claim 153, wherein said substantive characteristic of said content comprises at least one of objectionable subject matter contained in said requested content, language of text contained in said requested content, security-sensitive information contained in said requested content, premium subject matter contained in said requested content, or a user-identified type of subject matter contained in said requested content.

30